

WHAT IS CLAIMED IS:

1. A system, comprising:

5 a plurality of subsystems configured to execute within a virtual machine on the system, wherein two or more of the subsystems each provide a version of an isomorphic interface to functions of the subsystem; and

a proxy mechanism configured to generate a proxy to one of the two or more
10 subsystems that provides a correct version of the isomorphic interface for one of the plurality of subsystems at runtime of the one of the plurality of subsystems, wherein the proxy is configured to:

receive a call to the isomorphic interface from the one of the plurality of
15 subsystems;

convert the call in accordance with the version of the isomorphic interface
provided by the one of the two or more subsystems; and

20 forward the converted call to the one of the two or more subsystems for execution.

2. The system as recited in claim 1, wherein the proxy mechanism is further
configured to generate another proxy to another one of the two or more subsystems that
25 provides a different version of the isomorphic interface for another one of the plurality of subsystems at runtime of the other one of the plurality of subsystems.

3. The system as recited in claim 1, wherein, to generate a proxy to one of the two or
more subsystems that provides a correct version of the isomorphic interface for one of the
30 plurality of subsystems at runtime of the one of the plurality of subsystems, the proxy

mechanism is further configured to determine the one of the two or more subsystems that provides the correct version of the isomorphic interface for the one of the plurality of subsystems at runtime.

5 4. The system as recited in claim 3, wherein the one of the plurality of subsystems is configured to specify the isomorphic interface to be proxied to the proxy mechanism, and wherein the proxy mechanism is further configured to determine the one of the two or more subsystems that provides the correct version of the isomorphic interface and to generate the proxy according to said specification.

10

5. The system as recited in claim 1, wherein the proxy mechanism is further configured to generate another proxy configured to return results of said execution from the one of the two or more subsystems to the one of the plurality of subsystems.

15 6. The system as recited in claim 1, wherein the proxy mechanism is further configured to generate a proxy to one of the two or more subsystems that provides a correct version of the isomorphic interface for one of the plurality of subsystems at runtime of the one of the plurality of subsystems in response to said call to the isomorphic interface from the one of the plurality of subsystems.

20

7. The system as recited in claim 1, wherein the proxy mechanism is further configured to provide an API to the subsystems, wherein the API is configured for use by the subsystems to specify isomorphic interfaces to be proxied to the proxy mechanism.

25 8. The system as recited in claim 1, wherein the proxy is further configured to convert the call in accordance with the version of the isomorphic interface provided by the one of the two or more subsystems using Java Reflection.

9. The system as recited in claim 1, wherein the virtual machine is a Java Virtual
30 Machine (JVM).

10. The system as recited in claim 1, wherein the one of the plurality of subsystems is an application, and wherein the two or more subsystems are versions of a runtime library.

5 11. The system as recited in claim 1, wherein the one of the plurality of subsystems and the two or more subsystems are applications.

12. The system as recited in claim 1, wherein the plurality of subsystems are mobile agents.

10

13. A system, comprising:

15 a plurality of subsystems configured to execute within a virtual machine on the system; and

a proxy mechanism configured to:

20 generate a proxy for a version of an interface between two of the plurality of subsystems at runtime, wherein the proxy is configured to convert calls between the two subsystems in accordance with the version of the interface; and

25 generate another proxy for another version of the interface between another two of the plurality of subsystems at runtime, wherein the other proxy is configured to convert calls between the other two of the plurality of subsystems in accordance with the other version of the interface.

30 14. The system as recited in claim 13, wherein the proxy is configured to:

receive from a first of the two subsystems a call to a second of the two subsystems;

5 convert the call in accordance with the version of the interface; and

 forward the converted call to the second subsystem for execution by the second subsystem.

10 15. The system as recited in claim 14, wherein the proxy mechanism is further configured to generate another proxy configured to return results of said execution from the second subsystem to the first subsystem.

15 16. The system as recited in claim 14, wherein the proxy mechanism is further configured to generate the proxy for the version of the interface between the two subsystems in response to said call to the second subsystem.

17. The system as recited in claim 14, wherein the proxies are configured to convert the calls between the subsystems using Java Reflection.

20

18. The system as recited in claim 13, wherein a first of the two subsystems is configured to specify the interface to be proxied to the proxy mechanism, and wherein the proxy mechanism is further configured to generate the proxy for the interface between the two subsystems in accordance with said specification.

25

19. The system as recited in claim 13, wherein the virtual machine is a Java Virtual Machine (JVM).

30 20. A system, comprising:

a processor; and

a memory comprising program instructions, wherein the program instructions are
5 executable by the processor to implement a proxy mechanism configured
to generate a proxy to one of a plurality of subsystems that provide
versions of an isomorphic interface for another subsystem at runtime of the
other subsystem;

10 wherein the proxy is configured to:

receive a call to the isomorphic interface from the other subsystem;

convert the call in accordance with the version of the isomorphic interface
15 provided by the one of the plurality of subsystems; and

forward the converted call to the one of the plurality of subsystems for
execution.

20 21. The system as recited in claim 20, wherein the proxy mechanism is further
configured generate another proxy to another one of the plurality of subsystems that
provides a different version of the isomorphic interface for a different subsystem at
runtime of the different subsystem.

25 22. The system as recited in claim 20, wherein the proxy mechanism is further
configured to generate another proxy configured to return results of said execution from
the one of the plurality of subsystems to the other subsystem.

23. The system as recited in claim 20, wherein other subsystem is configured to
30 specify the isomorphic interface to be proxied to the proxy mechanism, and wherein the

proxy mechanism is further configured to generate the proxy according to said specification.

24. The system as recited in claim 20, wherein the program instructions are executable by the processor to implement a virtual machine in the memory, wherein the proxy mechanism, the subsystems, and the proxy are configured to operate within the virtual machine.

25. The system as recited in claim 20, wherein the virtual machine is a Java Virtual Machine (JVM).

26. The system as recited in claim 20, wherein the proxy is further configured to convert the call in accordance with the version of the isomorphic interface provided by the one of the plurality of subsystems using Java Reflection.

27. A system, comprising:

means for generating proxies to versions of an isomorphic interface provided by two or more subsystems for one or more other subsystems in a virtual machine at runtime;

means for the proxies to convert calls from the one or more other subsystems to the versions of the isomorphic interface provided by the two or more subsystems; and

means for the proxies to forward the converted calls to the two or more subsystems for execution.

28. A method, comprising:
- a proxy mechanism generating, for a subsystem, a proxy to one of a plurality of subsystems that provide versions of an isomorphic interface at runtime of the subsystem;
- the proxy receiving a call to the isomorphic interface from the subsystem;
- the proxy converting the call in accordance with a version of the isomorphic interface provided by the one of the plurality of subsystems; and
- the proxy forwarding the converted call to the one of the plurality of subsystems for execution.
29. The method as recited in claim 28, further comprising the proxy mechanism generating another proxy to another one of the plurality of subsystems that provides a different version of the isomorphic interface for another subsystem at runtime of the other subsystem.
30. The method as recited in claim 28, further comprising:
- the proxy mechanism generating another proxy configured to return results of executing the call from the one of the plurality of subsystems to the subsystem;
- the one of the plurality of subsystems executing the call; and
- the other proxy returning results of said executing the call to the subsystem.
31. The method as recited in claim 28, further comprising the subsystem specifying

the isomorphic interface to be proxied to the proxy mechanism, wherein the proxy mechanism generates the proxy to the one of the plurality of subsystems according to said specification.

5 32. The method as recited in claim 28, further comprising the proxy mechanism receiving the call to the isomorphic interface, wherein the proxy mechanism generates the proxy to the one of the plurality of subsystems in response to said receiving the call.

10 33. The method as recited in claim 28, wherein the proxy converts the call using Java Reflection.

34. The method as recited in claim 28, wherein the virtual machine is a Java Virtual Machine (JVM).

15 35. The method as recited in claim 28, wherein the subsystem is an application, and wherein the plurality of subsystems are versions of a runtime library.

36. The method as recited in claim 28, wherein the subsystem and the plurality of subsystems are applications.

20

37. The method as recited in claim 28, wherein the subsystem and the plurality of subsystems are mobile agents.

25 38. A method, comprising:

generating a proxy for a version of an interface between two of a plurality of subsystems in a virtual machine at runtime, wherein the proxy is configured to convert calls between the two subsystems in accordance with the version of the interface; and

30

generating another proxy for another version of the interface between another two
of the plurality of subsystems at runtime, wherein the other proxy is
configured to convert calls between the other two of the plurality of
subsystems in accordance with the other version of the interface.

39. The method as recited in claim 38, further comprising:

the proxy receiving from a first of the two subsystems a call to a second of the two
subsystems;

the proxy converting the call in accordance with the version of the interface; and

the proxy forwarding the converted call to the second subsystem for execution by
the second subsystem.

40. The method as recited in claim 39, wherein the proxy converts the call in
accordance with the version of the interface using Java Reflection.

41. The method as recited in claim 39, further comprising generating another proxy
configured to return results of said execution from the second subsystem to the first
subsystem.

42. The method as recited in claim 39, further comprising generating the proxy for the
version of the interface between the two subsystems in response to said call to the second
subsystem.

43. The method as recited in claim 38, further comprising:

a first of the two subsystems specifying the interface to be proxied; and

generating the proxy for the interface between the two subsystems in accordance with said specification.

5 44. The method as recited in claim 38, wherein the virtual machine is a Java Virtual Machine (JVM).

45. A computer-accessible medium comprising program instructions, wherein the
10 program instructions are configured to implement:

a proxy mechanism generating, for a subsystem, a proxy to one of a plurality of
subsystems that provide versions of an isomorphic interface at runtime of
the subsystem;

15

the proxy receiving a call to the isomorphic interface from the subsystem;

the proxy converting the call in accordance with a version of the isomorphic
interface provided by the one of the plurality of subsystems; and

20

the proxy forwarding the converted call to the one of the plurality of subsystems
for execution.

46. The computer-accessible medium as recited in claim 45, wherein the program
25 instructions are further configured to implement the proxy mechanism generating another
proxy to another one of the plurality of subsystems that provides a different version of the
isomorphic interface for another subsystem at runtime of the other subsystem.

47. The computer-accessible medium as recited in claim 45, wherein the program
30 instructions are further configured to implement:

the proxy mechanism generating another proxy configured to return results of
executing the call from the one of the plurality of subsystems to the
subsystem;

5

the one of the plurality of subsystems executing the call; and

the other proxy returning results of said executing the call to the subsystem.

10 48. The computer-accessible medium as recited in claim 45, wherein the program
instructions are further configured to implement the subsystem specifying the isomorphic
interface to be proxied to the proxy mechanism, wherein the proxy mechanism generates
the proxy to the one of the plurality of subsystems according to said specification.

15 49. The computer-accessible medium as recited in claim 45, wherein the program
instructions are further configured to implement the proxy mechanism receiving the call
to the isomorphic interface, wherein the proxy mechanism generates the proxy to the one
of the plurality of subsystems in response to said receiving the call.

20 50. The computer-accessible medium as recited in claim 45, wherein the proxy
converts the call using Java Reflection.

51. The computer-accessible medium as recited in claim 45, wherein the virtual
machine is a Java Virtual Machine (JVM).

25

52. The computer-accessible medium as recited in claim 45, wherein the subsystem is
an application, and wherein the plurality of subsystems are versions of a runtime library.

53. The computer-accessible medium as recited in claim 45, wherein the subsystem
30 and the plurality of subsystems are applications.

54. The computer-accessible medium as recited in claim 45, wherein the subsystem and the plurality of subsystems are mobile agents.

5

55. A computer-accessible medium comprising program instructions, wherein the program instructions are configured to implement:

generating a proxy for a version of an interface between two of a plurality of
10 subsystems in a virtual machine at runtime, wherein the proxy is
configured to convert calls between the two subsystems in accordance
with the version of the interface; and

generating another proxy for another version of the interface between another two
15 of the plurality of subsystems at runtime, wherein the other proxy is
configured to convert calls between the other two of the plurality of
subsystems in accordance with the other version of the interface.

56. The computer-accessible medium as recited in claim 55, wherein the program
20 instructions are further configured to implement:

the proxy receiving from a first of the two subsystems a call to a second of the two
subsystems;

25 the proxy converting the call in accordance with the version of the interface; and

the proxy forwarding the converted call to the second subsystem for execution by
the second subsystem.

30 57. The computer-accessible medium as recited in claim 56, wherein the proxy

converts the call in accordance with the version of the interface using Java Reflection.

58. The computer-accessible medium as recited in claim 56, wherein the program instructions are further configured to implement generating another proxy configured to return results of said execution from the second subsystem to the first subsystem.

59. The computer-accessible medium as recited in claim 56, wherein the program instructions are further configured to implement generating the proxy for the version of the interface between the two subsystems in response to said call to the second subsystem.

60. The computer-accessible medium as recited in claim 55, wherein the program instructions are further configured to implement:

a first of the two subsystems specifying the interface to be proxied; and

generating the proxy for the interface between the two subsystems in accordance with said specification.

61. The computer-accessible medium as recited in claim 55, wherein the virtual machine is a Java Virtual Machine (JVM).